

## **Multiprocessor system with microprogrammed means for partitioning the processes between the processors**

Patent number: JP5250337  
Publication date: 1993-09-28  
Inventor: JIYORUJIYU RUKURUTEIE  
Applicant: BULL SA  
Classification:  
- international: G06F15/16  
- european: G06F9/46C4S  
Application number: JP19920292966 19921030  
Priority number(s): FR19910013431 19911030

[View INPADOC patent family](#)

Also Published : EP0540383 (A1);FR2683344 (A1);EP0540383 (B1)

### **Abstract of JP5250337**

**PURPOSE:** To provide a multiprocessor having microprogrammed means for dispatching processing to processor. **CONSTITUTION:** In this multiprocessor system, an event to attain the dispatching change of processing to a CPU and a CPU<sub>i</sub> starts the execution of a dispatching software in charge of the definition of new dispatching as an event or the function of various data. When the number of processors CPU and CPU<sub>i</sub> is large, it is necessary to optimize a microsoftware. Therefore, the microsoftware is divided into plural dedicated microprogram modules for deciding dispatching in response to the respective categories of specified events. The central microprogram module considers the event except for the processing due to the dedicated modules.

Claims of corresponding document: **EP0540383**

1. Système informatique multiprocesseur dont une pluralité de processeurs (CPU) sont munis chacun de moyens microprogrammés (CP) pour effectuer la répartition des processus (JP) aux processeurs (CPU) du système, lesdits moyens microprogrammés (CP) étant activés dans un processeur (CPU) en réponse à tout événement détecté par ledit processeur (CPU) et qui est susceptible de provoquer une modification dans le choix des processus (JP) devant être exécutés et des processeurs (CPU) qui les exécutent, ledit système étant caractérisé en ce que lesdits moyens microprogrammés (CP) comportent au moins un module spécialisé de microprogramme (CHSEL, DQSEL, EQSEL) capable de déterminer ladite répartition en réponse à une catégorie d'événements particuliers et un module central de microprogramme (CTSEL) capable de déterminer ladite répartition pour des événements autres que ceux traités par lesdits modules spécialisés (CHSEL, DQSEL, EQSEL).

2. Système selon la revendication 1 caractérisé en ce que chaque catégorie d'événements particuliers comprend des événements parmi ceux qui ont la plus forte probabilité de se produire au cours de l'exploitation du système.

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平5-250337

(43) 公開日 平成5年(1993)9月28日

(51) Int.Cl.<sup>5</sup>

識別記号

庁内整理番号

F I

技術表示箇所

G 0 6 F 15/16

3 8 0 Z 9190-5L

審査請求 有 請求項の数10(全 17 頁)

(21) 出願番号 特願平4-292966

(22) 出願日 平成4年(1992)10月30日

(31) 優先権主張番号 9 1 1 3 4 3 1

(32) 優先日 1991年10月30日

(33) 優先権主張国 フランス (F R)

(71) 出願人 390035633

ブル・エス・アー

フランス国、75116・パリ、アブニユ・ド  
ウ・マラコフ、121

(72) 発明者 ジョルジュ・ルクルティエ

フランス国、78000・ベルサイユ、リュ・  
シヤンールガルドウ、33・ビス

(74) 代理人 弁理士 川口 義雄 (外2名)

(54) 【発明の名称】 処理をプロセッサにディスパッチするためのマイクロプログラム手段を有するマルチプロセッサシステム

(57) 【要約】

【目的】 処理をプロセッサにディスパッチするためのマイクロプログラム手段を有するマルチプロセッサを提供する。

【構成】 マルチプロセッサシステムにおいて、処理のプロセッサ (CPU, CPU i) へのディスパッチングを変更し得る事象は、事象又は種々のデータの関数として新たなディスパッチングを定義することを受け持つディスパッチングソフトウェアの実行を開始する。プロセッサ (CPU, CPU i) の数が大きいときには、マイクロソフトウェアは最適化されねばならない。このために、マイクロソフトウェアは、特定の事象のそれぞれのカテゴリにตอบสนองしてディスパッチングを決定するための複数の専用マイクロプログラムモジュールに細分される。中央マイクロプログラムモジュールは、専用モジュールによって処理される以外の事象を考慮する。

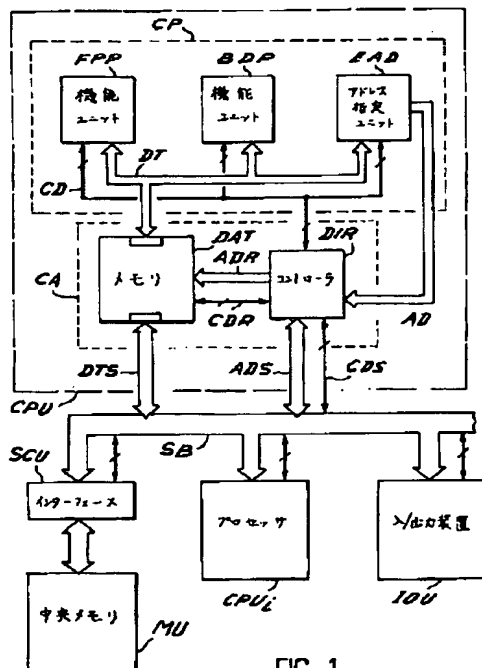


FIG. 1

## 【特許請求の範囲】

【請求項1】 複数のプロセッサの各々に、システムのプロセッサへの処理のディスパッチングを実行するためのマイクロプログラム手段が備えられているマルチプロセッサ情報処理システムであって、前記マイクロプログラム手段がプロセッサ内で、前記プロセッサによって検出され、且つ実行される処理及び該処理を実行するプロセッサの選択に変更を起こし得る任意の事象に応答して能動化され、更に前記マイクロプログラム手段が、特定の事象のカテゴリに該当して前記ディスパッチングを決定し得る少なくとも1つの専用マイクロプログラムモジュールと、前記専用モジュールによって処理される以外の事象に対して前記ディスパッチングを決定し得る中央マイクロプログラムモジュールとを含むことを特徴とするマルチプロセッサシステム。

【請求項2】 前記各特定事象のカテゴリが、特にシステムオペレーションの間に起こる確率が最も高い事象を含むことを特徴とする請求項1に記載のシステム。

【請求項3】 前記各特定事象のカテゴリが、実行される処理及び該処理を実行するプロセッサの選択に単純な変更を起こす事象を含むことを特徴とする請求項1または2に記載のシステム。

【請求項4】 前記システムにおいて実行され得る処理のアイデンティティが、事象待ち処理キューまたは少なくとも1つのレディ状態処理キュー、即ちプロセッサ内でアクティブなもしくはプロセッサ待ち中の処理キュー内に記憶されており、前記事象のカテゴリが、それぞれレディ状態処理キューにおける単一処理の追加、抹消または移動を惹起し得る事象を含むことを特徴とする請求項3に記載のシステム。

【請求項5】 前記レディ状態処理キューが、一度にただ1つのプロセッサのみが前記レディ状態処理キューの状態を変更し得ることを保証するために前記マイクロプログラム手段によって使用されるロックと関係していることを特徴とする請求項4に記載のシステム。

【請求項6】 前記キューが、一度にただ1つのプロセッサのみが前記待ち状態処理キューの状態を変更し得ることを保証するために前記マイクロプログラム手段によって使用される少なくとも1つのロックと関係していることを特徴とする請求項4または5に記載のシステム。

【請求項7】 前記専用モジュールが、前記中央モジュールを呼び出せるように設けられていることを特徴とする請求項1から6のいずれか一項に記載のシステム。

【請求項8】 実行されるべき新たな処理のアイデンティティを他のプロセッサに通知するリクエストを発信するために、プロセッサの前記マイクロプログラム手段によって適宜使用される処理間対話手段を含んでおり、前記対話手段が、一度にただ1つのプロセッサのみが前記リクエストを発信することを保証するために前記マイクロプログラム手段によって使用される第1の対話ロック

と関係しており、前記各専用モジュールが、該専用モジュールの実行が対話を含む一方前記ロックがかけられている場合には、その実行を停止し且つ前記中央モジュールを呼び出すように設計されていることを特徴とする請求項7に記載のシステム。

【請求項9】 前記対話手段が、各プロセッサと関係しており且つ前記リクエストを送信するために前記マイクロプログラム手段によって使用されるメールボックスを含んでおり、前記メールボックスの各々が、リクエストを発信したプロセッサによって占有状態にされ且つリクエストが割り当てられたプロセッサによってリクエスト実行後に解除されるロックと関係しており、前記各専用モジュールが、該専用モジュールの実行がリクエストをプロセッサに送ることを含む一方そのメールボックスのロックがかけられている場合には、その実行を停止し且つ前記中央モジュールを呼び出すように設計されていることを特徴とする請求項8に記載のシステム。

【請求項10】 制約条件を有する処理、即ちその実行がある特定のプロセッサによって行われるべき処理がシステム内に存在するかどうかをその状態が表わしている条件インジケータを含んでおり、前記マイクロプログラム手段が、前記インジケータが制約条件の存在を知らせているときには、処理のプロセッサへのディスパッチングが前記中央モジュールによって行われるように設計されていることを特徴とする請求項1から9のいずれか一項に記載のシステム。

## 【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明はマルチプロセッサタイプの情報処理システムの分野に属し、特にかかるシステムにおける処理のプロセッサへの動的タスク割当てを保証するために使用されるメカニズムに関する。

【0002】

【従来の技術】 処理をプロセッサに割当てること（通常は“ディスパッチング”と称される）は、特にトランザクションタイプのアプリケーションにおいては、その用法がシステムパフォーマンスに大きく影響するオペレーションである。パフォーマンスに及ぼすディスパッチングの影響は、ディスパッチングをトリガし得る事象の平均頻度はプロセッサの数に比例して増大するが故に、システムプロセッサの数が大きくなると特に変動し易い。ディスパッチングは、実行される処理の選択に変更を起こし得る任意の事象によってトリガされるオペレーションである。大抵の場合、かかる事象は、処理間でオペレーションを同期化することから生じ、処理とは、アプリケーションプログラム、または入/出力プログラムのようなオペレーティングシステムプログラムを実行するものであり得る。例えばプロセッサの状態の変化、または処理をシステムに導入もしくはシステムから消去したり処理の1つのプライオリティレベルを変更する特定の命

令に関係する他のタイプの事象もあり得る。このような事象の出現は、事象及び、例えば実行される準備が整っている処理のプライオリティ順位を考慮するような種々の基準の関数として、処理をプロセッサに新たに割り当てる決定をまかされたシステムマイクロプログラムを能動化する効果を有する。

【0003】マイクロプログラム使用のケースでは、ディスパッチングは2つのフェーズで行われる。第1の選択フェーズは、“SELECT”と称される第1のマイクロプログラムモジュールの実行に対応し、実行されるべき処理及びその処理を実行するプロセッサの選択を行なうために与えられる。この選択フェーズには、関係するプロセッサに実行されるべき新たな処理を通知する役割を果たす割込みを送ることが含まれる。アドレス指定されたプロセッサによってかかる割込みが受信されると、実質的にその実行が割り込まれた処理の内容を保存し且つ実行されるべき新たな処理の内容をプロセッサ内にロードすることからなる実行フェーズが、該プロセッサ内でトリガされる。この実行フェーズは、“EXEC”と称されるマイクロプログラムモジュールを含むプロセッサによる並行処理に対応する。

【0004】ディスパッチングをトリガする事象に対するシステムの応答時間を向上するためには選択オペレーションが分散式に行われ得ることが望ましく、これには、システムの全てのプロセッサまたは少なくともそのうちの幾つかが各々が選択オペレーションを実行し得ることが必要である。この問題は、1986年5月20日発行米国特許第4,590,550号(欧州特許B130504号に対応)の課題であり、前記特許は参照により本明細書の一部を構成するものとする。この特許は特に、各々が選択オペレーションを実行しようとする複数のプロセッサ間に起こり得る対立に係わる問題を取り扱っている。選択オペレーションは、処理及びプロセッサの状態のようなシステム全体で同じ値を有し続けねばならないシステムデータを取扱うが故に、このような対立は無条件に回避されねばならない。このためには特にロックが与えられる。即ち、先行のディスパッチングが完全に終了しないうちに他のプロセッサが新たな選択フェーズを実行しないように、ディスパッチングを実行しているプロセッサによって能動化及び保持されるインジケータ手段が与えられる。

【0005】

【発明が解決しようとする課題】本発明の目的は、プロセッサの数が増大しても相対的なシステムパフォーマンスの低下を招くことのないように上述のディスパッチングの最適実施例を提案することである。

【0006】この最適化を求めて、マルチプロセッサシステムにおいて実行される種々のタイプのディスパッチングを統計的に分析した。トランザクション型アプリケーションにおいては、この分析によって約75%のケー

スでディスパッチングオペレーションは最終的に、実行する処理の選択に変更がないか、または単一処理のみの実行に割込み且つ一般的にはそれを別の処理で置き換える結果となることが確認された。従って上記のディスパッチングが頻繁なケースは、処理されるべき最も単純なものとして推定され、システムの一貫性が全ての環境下で保存され得るようにこの状況を利用する手段を見いだす必要があった。

【0007】第2に、選択オペレーションに関与する2つの資源カテゴリを区別することができる。第1のカテゴリは、一般的には処理キューによって物理的に実現される、システム内に設定された処理の状態に係わる情報を含む。処理キューは一般に事象と関係して存在する。各キューは、その実行が割り込まれて事象または関係メッセージの出現を待っている全ての処理のアイデンティティを含んでいる。一方で、実行中の処理及び実行する準備の整った処理、即ち使用可能なプロセッサがないために実行され得ない処理を含む少なくとも1つのキューが存在する。第2の資源カテゴリは、プロセッサ間の連絡を可能とする対話手段を含む。かかる手段は通常は、特に選択オペレーションを実行しているプロセッサが、アドレス指定されたプロセッサに、実行されるべき新たな処理を通知できるようにする、メールボックスと称されるデータ交換ゾーンに係る割込みメカニズムによって実現される。ディスパッチングが最も頻繁となる環境を更に詳細に分析することにより、上記資源のうち、幾つかのものだけが使用されることが極めて多いことが確認された。実際、ディスパッチングをトリガし得る事象が、実行中の処理による事象またはメッセージの通知からなるならば、2つのケースが起こり得る。第1のケースはその事象を待つ処理のないケースである。事象が発生したプロセッサは大部分のケースで、事象が発生したことを示すためにセマフォタイプのデータ構造を更新し得る。このプロセッサは、それまでにそれが実行していた処理を続けて実行することができ、システムのプロセッサに処理をディスパッチングすることにおいて何が起ころうと変更はない。その場合、セマフォに係るデータ構造のみが関与する。もう1つのケースは、少なくとも1つの処理がこの事象を待っているケースである。キュー内の最高プライオリティを有する処理がそのキューから抽出され、レディ状態処理キュー内に挿入され、この処理は、そのプライオリティレベルに従ってシステムのプロセッサの1つにおいてアクティブな処理と置き換わることができる。このオペレーションに関与する資源は、セマフォに係るデータ構造及び処理キュー、レディまたはアクティブ状態処理キュー、及び必要に応じて、待ち状態にあった処理が別のプロセッサにおいてアクティブ状態だった処理と置き換わらねばならないならば、プロセッサ間対話手段である。

【0008】別の例として、アクティブ状態処理が、そ

の実行を継続し得るために事象が発生したかまたはメッセージが存在するか問合せねばならない相補的なケースを例にとると、ここでも2つのケースに区別される。第1のケースでは、事象またはメッセージは既に通知されており、これは、セマフォに関係するデータ構造をアクセスすることによりプロセッサによって検証される。処理はその実行を続けることができる。第2のケースでは、事象またはメッセージは通知されておらず、処理の実行は割り込まねばならない。その場合、レディ状態処理キュー内の最高プライオリティを有する処理が、プロセッサによってそれまで実行されていた処理にとって替わる。資源の使用については、第1のケースではセマフォに関係するデータ構造のみが関与し、第2のケースではレディ状態処理キューも関与する。これとは反対に対話手段はいずれのケースでも使用されない。

【0009】特定の状況で上記実施例の幾つかに基づき、処理をプロセッサにディスパッチするオペレーションは、各々が特定の事象または状況の1つのカテゴリに対応していて、実際にはディスパッチングオペレーションに関与し得る資源セットの幾つかしか使用しない複数のより単純なオペレーションに分割することができる。

【0010】従って本発明の目的は、ディスパッチングにおける所定のフェーズを並行処理し得る可能性を増大しようという観点から上記知見を利用することである。しかしながら、提案される実施例は機能的に信頼性があり且つ使用が単純なものであらねばならない。

【0011】

【課題を解決するための手段】この目的は、ディスパッチングを行うソフトウェアを細分することにより達成され、この細分は、処理をプロセッサにディスパッチングする上で同じタイプの変更を起こる事象の1つのカテゴリまたはセットに属する少なくとも1つの専用モジュールを定義するように選択され、これは、同じタイプの資源が使用されることを意味する。複数のこのタイプのモジュールを与えることにより、システム資源の点で対立する危険性もなく複数のモジュールが同時に実行される可能性を増大する結果となる。

【0012】カテゴリに適合しない事象、即ち専用モジュールによって処理され得ない事象を考慮するための特別モジュールを与えることも適当である。

【0013】より正確には、本発明は、その複数のプロセッサの各々に、システムのプロセッサへの処理のディスパッチングを実行するためのマイクロプログラム手段が備えられているマルチプロセッサ情報処理システムであって、前記マイクロプログラム手段がプロセッサ内で、前記プロセッサによって検出され、且つ実行される処理及び該処理を実行するプロセッサの選択に変更を起こし得る任意の事象にตอบสนองして能動化され、更に前記マイクロプログラム手段が、特定の事象のカテゴリにตอบสนองして前記ディスパッチングを決定し得る少なくとも1つ

の専用マイクロプログラムモジュールと、前記専用モジュールによって処理される以外の事象に対して前記ディスパッチングを決定し得る中央マイクロプログラムモジュールとを含むことを特徴とするマルチプロセッサシステムを提供する。

【0014】システムパフォーマンスの見地から本発明をよりよく利用するため、本発明の別の特徴においては、特定事象のカテゴリは、特にシステムオペレーションの間に起こる確率が最も高い事象を含む。

【0015】並行処理の可能性を増強するため、本発明の別の特徴においては、特定事象の各カテゴリは、実行される処理及び該処理を実行するプロセッサの選択に単純な変更を起こす事象を含む。

【0016】本発明は更に、前記システムにおいて実行され得る処理の状態が、事象待ち処理キューまたはレディ状態処理キュー、即ちプロセッサ内でアクティブなもしくはプロセッサ待ち中の処理キュー内に記憶されているならば、事象のカテゴリが、レディ状態処理キューにおいてそれぞれ単一処理JPの追加、木梢または移動を惹起し得る事象を含む、前記システムの前記の実施例をも提供する。

【0017】専用モジュールは、関係事象に完全に対応して且つ全ての環境下でディスパッチングを実行するように設計することができる。しかしながらそれには、モジュールをより複雑にしたりマイクロプログラムのメモリサイズを増大する必要がある。この欠点を解消するため本発明の変形例によれば、好ましくは複数のプロセッサにおける同時ディスパッチングオペレーション間に対立が存在するような例外的複合状況に対応する、所定のケース外での選択オペレーションを完全に実行するように各専用モジュールは与えられている。各専用モジュールは、上記ケースを検出して中央モジュールを呼び出すように与えられている。従って中央モジュールは、専用モジュールによって処理されない上記全ての例外的ケースを統括的に処理するように設計されている。

【0018】従って特定の実施例においては本発明のシステムは、実行されるべき新たな処理NJ Pのアイデンティティを他のプロセッサに通知するリクエストを発信するために、プロセッサのマイクロプログラム手段によって適宜使用される処理間対話手段を含んでいる。一方でかかる対話手段は、一度にただ1つのプロセッサのみがリクエストを発信することを保証するためにマイクロプログラム手段によって使用される第1の対話ロックと関係している。最後に各専用モジュールは、該専用モジュールの実行が対話を含む一方、前記ロックがかけられている場合には、その実行を停止し且つ中央モジュールを呼び出すように設計されている。

【0019】

【実施例】実施例の更なる態様及び詳細を以下の説明において詳述する。

【0020】図1に示したシステムは、情報処理システムの中央サブシステムと称されるものを構成している。これは、相互に並びに中央メモリMU及び入/出力装置IOUと通信し得るようにシステムバスSBに接続されている複数のプロセッサCPU、CPUiで構成されている。入/出力装置IOUによって中央サブシステムは周辺サブシステム（図示なし）と通信することができる。

【0021】中央メモリMUは、メモリコントローラ及びバスSBのコントローラの両方の役割を果たすインターフェース回路SCUを介してシステムバスSBに接続されている複数のメモリモジュールまたはカードによって実現することができる。通常の方法ではバスSBは、データバスDTS、アドレスバスADS及びコマンド及びコントロールラインCDSによって構成される。

【0022】各プロセッサCPUは必然的に、マイクロプログラム処理手段CPと、システムバスSBとのインターフェースとして作用するキャッシュメモリCAとを含んでいる。キャッシュメモリCAは、データバスDTSに接続されたメモリ回路DATと、アドレスバスADS及びコントロールラインCDSに接続されたコントローラDIRとで構成されている。処理手段CPは、それぞれ特定機能に割り当てられた複数の機能ユニットEAD、BDP、FPPを含んでいる。特にユニットEADは、アドレスラインADを介してコントローラDIRに接続されているアドレス指定ユニットである。ユニットEAD、BDP、FPPは内部データバスDTを介してメモリ回路DATにも、また内部コマンド及びコントロールラインCDを介してコントローラDIRにも接続されている。

【0023】コントローラDIRはそれぞれアドレスラインADR及びコントロールラインCDRを介してメモリ回路DATの読取り及び書込みオペレーションを制御する。更にコントローラDIRは、中央メモリMUとメモリ回路DATとの間のデータ転送を行なうために、中央メモリMUの読取り及び書込みオペレーションをも制御する。最後に、図示した実施例においてはコントローラDIRは更に、プロセッサCPUと他のプロセッサCPUiとの間または入/出力装置IOUとのメッセージ交換（例えばリクエスト及び肯定応答）の目的を果たすインターフェース回路をも含んでいる。

【0024】プロセッサCPUの機能ユニットの詳細実施例は、“Processeurs a plusieurs unites de traitement microprogrammees” [Processor with a Plurality of Microprogrammed Processing Units] の標題で1991年6月26日公開の欧州特許出願公開第434 483号に記載されている。

【0025】更に、システムのプロセッサCPU、CP

UI間の対話を可能とする手段は、“Procede de dialogue entre les processeurs d'un systeme, systeme pour samise en ouvre et utilisation pour la repartition des processus aux processeurs” [Methods for Dialog Among the Processors of a System, System for Employing it, and Use for Dispatching Processes to Processors] の標題で1990年6月5日出願の仏国特許出願第90. 06948号に従って実現することができる。

【0026】図1のシステムの一般機能については、参照により本明細書の一部を構成するものとする上記2つの特許出願を参照することが適当である。

【0027】この点において本発明は、システムマイクロソフトウェアの一部である特定のマイクロプログラム手段によって実施することができる。かかるマイクロプログラムは、システムによって考慮され得る事象を表わす所定数のインジケータの関数として、実行されるべきモジュールを選択する役割を果たす“割り込みマネージャ”と称されるマイクロソフトウェアモジュールと協働するように与えられている。本発明の実施を可能とする特定のマイクロプログラムを図8～図11と合わせて説明するが、まず最初にかかるマイクロプログラムによって操作されるシステム資源について特定の説明を行なう。

【0028】図2は、システム内に設定された処理が関係する“処理制御ブロック”PCBとして公知のデータ構造を示す。このブロックは、処理に係わると共にシステムにとって有効な全ての情報を含む所定数の32ビットワードによって構成されているメモリゾーンである。図には本発明を実施するのに係わる情報のみを示してある。この情報のなかでワードPMW0内には、処理のプライオリティレベルを表わすフィールドPRIと処理の状態を示すフィールドSTATEとが認められる。このフィールドSTATEがとる値は特に、処理がアクティブであるのか、またはレディ状態で空きプロセッサを待っているのか、または事象を待っているのかを示す。処理がセマフォに関係する事象を待っているときにはワードPMW2はセマフォのアドレスに割り当てられる。ワードPMW3のフィールドCPSMは、その2進分布が、処理を実行するよう許可されたシステムプロセッサを規定しているマスクである。“ゾーンレジスタ”と称されるワード群は、処理の内容、即ちその実行が割り込まれる直前に最後に処理を実行したプロセッサのレジスタの内容を保存する役割を果たす。

【0029】図3は、“処理制御ブロック”CPCBと

称され、送り側のプロセッサとそのブロックに関する受け側のプロセッサとの間の情報交換のためのメールボックスとして作用する、各プロセッサに関するメモリゾーンを示す。

【0030】ブロックCPCBは、プロセッサ番号を含むフィールドCPNと、プロセッサ状態を示すフィールドCPSと、該プロセッサ内で実行されている処理の数を含むフィールドCJPと、レディ状態処理キューQ/PR/RDY内のこの処理のリンクのアドレスを表わすフィールドCPLDと、プロセッサが実行する予定の新たな処理の番号を示すフィールドNJPと、その処理のリンクのアドレスを示すフィールドNPLDとを含んでいる。

【0031】ブロックCPCBは更に、ロックとして作用し且つその論理状態が、メールボックスが占有されているか否かを知らせるデジタルインジケータDCLKをも含んでいる。

【0032】次に、処理キューを規定し得る資源を説明する。かかるエレメントの全ては既に、前出の米国特許第4,590,555号及び“semaphore device for computer”の標題の1983年7月26日発行米国特許第4,395,797号に詳細に説明されている。ここでは、本発明に関与する不可欠なエレメントのみを簡単に想起する。

【0033】既に述べたように、2つの処理キューカテゴリ、即ちレディ状態処理キューQ/PR/RDYと事象待ち処理キューとは区別される。これらのキューは、“リンク”として公知のデータ構造セットによって物理的に実現され、各リンクは1つの特定の処理と関係する。

【0034】図4は、このようなリンクPLのためのフォーマット例を示す。このリンクは32ビットワードで構成されており、下記の意味を有する複数のフィールドを含んでいる：

- NLは、キュー内の次のリンクのアドレスを表わすポインタであり、
- JPは、該リンクに関係する処理のアイデンティティであり、
- Rは、関係する処理の状態を表わすビットであり、
- Dは、ディスパッチングの実行を容易にするためのロックDCLKのコピーである補助ビットであり、
- TENは、処理を実行しているプロセッサまたは最後に処理を実行したプロセッサのアイデンティティであり、
- PRIは、処理のプライオリティレベルに対応する。

【0035】キューへのアクセスは、キュー内の最初のリンク、即ちキュー内の最も古い処理に関係するリンクのアドレスを表わすポインタによって行なうことができる。リンクがキューから消去されると、ポインタは、このリンクのフィールドNL内に含まれる値を取る。リン

クが加えられると、最後のリンクのフィールドNLは、この新たなリンクのアドレスに対応する値を取る。

【0036】本発明に関連して使用される特定の実施例によれば、レディでアクティブな処理とレディで非アクティブな処理とを合わせた単一キューQ/PR/RDYが与えられる。ビットRの論理値によって、処理がアクティブである(R=1)かそうでない(R=0)かが区別され得る。

【0037】事象待ち処理についても、そのリンクが図4に示したフォーマットを有するキューが使用される。各キューQ/PR/Sは“セマフォ”として公知のデータ構造に関係しており、各セマフォは特定のタイプの事象と関係している。

【0038】所定のセマフォが、図6に示したようなメッセージリンクによって物理的に実現されるメッセージキューQ/M/Sと関係するようにもできる。

【0039】セマフォSEMのフォーマットを図5に示す。

【0040】セマフォSEMは、下記の意味を有する複数のフィールドを含む2つの32ビットワードで形成されている：

- STAGは、メッセージキューに関係し得るか否かに従うセマフォのタイプを示しており、
- SMCは、関係キューが含み得る処理またはメッセージの最大数を示しており、
- SCTは、その絶対値がキュー内に含まれる処理またはメッセージの数を示し且つその符号が処理であるかメッセージであるかを示す代数的な値であり、
- PQHP/MQHPは、キュー内の最初の処理またはメッセージリンクのアドレスを表わすポインタであり、
- MQTPは、キュー内の最後の処理またはメッセージリンクのアドレスを表わすポインタである。

【0041】メッセージリンクMLのフォーマットを図6に示す。メッセージリンクMLは下記の意味を有する複数のフィールドを含んでいる：

- NLは、次のメッセージリンクのアドレスを示すポインタであり、
- SENDERは、メッセージの送り手である処理のアイデンティティであり、
- MPLは、メッセージのプライオリティレベルに対応しており、
- MTAGは、メッセージのタイプを規定するインジケータであり、
- MESSAGEは、メッセージ自体のために確保されているゾーンである。

【0042】空きメッセージリンク待ち処理キューまたは空きメッセージリンクキューの管理を可能にする特定のセマフォ及び空き処理リンクキューを管理するセマフォもある。

【0043】米国特許第4,395,757号は、処理

11

を同期化するためにセマフォがどのように使用されるかを記述している。従って、ここではかかるオペレーションを詳細に記述することはしないが、それらは、最初はゼロであるセマフォのカウントSCTをそれぞれ増分または減分する効果を有するPまたはVタイプ命令によって行われる。正のSCTカウントは、少なくとも1つの処理がセマフォに関係する事象またはメッセージを通知したことを意味する。これとは反対に負のSCTカウントは、少なくとも1つの処理がセマフォと関係する事象またはメッセージを待っていることを意味する。本発明においてP及びV命令がどのように処理されるかを、図8～図11を参照して後述する。

【0044】図7は、システムにとってその管理に有効な全てのデータをまとめる“システム制御ブロック”SCBと称される異なるデータ構造を示す。特にSCBは、ディスパッチングオペレーション時点で使用される特定のフィールドを含んでいる。それらのフィールドは下記の意味を有する：

—RDYLKは、レディ状態処理キューQ/PR/RDYと関係するロックであり、

—CSTは、特定のプロセッサによって実行される予定のシステム内に設定された処理の数を示す条件カウンタの値を含んでおり、

—MPDLは、プロセッサ間の対話に対するロックであり、

—SEMLKはそれぞれセマフォまたはセマフォ群に関係するロック群であり、

—VLD-CPU-MSKは、そのデジタル分布によってシステムの有効プロセッサを決定し得るマスクであり、

—IDLE-CPU-MSKは、そのデジタル分布によってシステム内の非アクティブプロセッサを決定し得るマスクである。

【0045】次に、2つの特定の実施例、即ちセマフォに対するP及びVタイプ命令において本発明をどのように実施し得るかを示す。

【0046】図8のフローチャートは、プロセッサCPUによって実行されたPタイプ命令によってトリガされるオペレーションを示す。プロセッサにおけるP命令の出現とは、この命令を含む処理は、特定のセマフォに関係する事象が発生したかまたはメッセージが送られてきたかどうかを検証しないと、その実行を続行できないということを想起されたい。命令Pを実行することは、次に説明する対応のマイクロプログラムを実行することである。。

【0047】ステップ1は、命令Pにおいて識別されたセマフォと関係するロックSEMLKの“TEST AND SET”オペレーションである。これ自体は公知のオペレーションは、システム制御ブロックSCB内に含まれているロックSEMLKをアクセスすることから

12

なる。ロックが解除されているならば（SEMLK=0）、ロックを占有状態にし（SEMLK:=1）、マイクロプログラムを続行する。これとは反対にロックがかけられていれば（SEMLK=1）、マイクロプログラムは、ロックが解除されるまでスタンバイを続ける。

【0048】次のステップ2は、セマフォSEMをアクセスし、その内容を分析し、それを更新する処理を行なうことからなるが、更新処理は特にカウントSCTを1単位だけ減分することからなる。カウントSCTが正であったならば、これは、処理によって要求された事象またはメッセージが既に通知されたことを意味している。その場合にはステップ3においてロックSEMLKが解除され（SEMLK:=0）、処理の次の命令を通常に実行することができる。セマフォがメッセージを含むタイプのものであるならば、セマフォの更新は、1つのメッセージリンクをメッセージキューから取り出すこととなることに留意されたい。更に、メッセージの内容はプロセッサの作業用レジスタ内にロードされる。特定のケースでは、オペレーションは空きメッセージリンク待ち処理の解放を行なうことができ、この場合にはマイクロプログラムは、この状況を取り扱う専用モジュールを呼び出す。

【0049】カウントSCTが負またはゼロであったならば、処理の実行は割込まねばならない。ステップ4においてロックRDYLKのTEST AND SETオペレーションが行われ、次いでステップ5において、実行中の処理リンクCJPがQ/PR/RDYからセマフォSEM関係キューQ/PR/Sに移される。ステップ6においてロックSEMLKは解除され、次いでステップ7においてブロックSCBのフィールドCSTがテストされる。もしCSTがゼロではないならば、これは、条件付き処理が存在することを意味する。そうするとマイクロプログラムは、ステップ8においてロックRDYLKを解除した後中央モジュールCTSELに飛ぶ。モジュールCTSELは、前出の仏国特許出願第90.06948号に従って実現することができる。

【0050】カウンタCSTがゼロであるならば、マイクロプログラムは、レディ状態処理キューQ/PR/RDYから処理を取り出すケースでは新たな処理NJPを選択するため、専用モジュールDQSELを呼び出す。

【0051】モジュールDQSELの実行はステップ9から開始され、ステップ9は、レディ状態処理キューQ/PR/RDYをアクセスし且つそれを分析する処理を行なう。この分析は実質的に、キューQ/PR/RDY（空ではないと仮定する）内のレディ状態処理NJPのうちアクティブ状態に変更されねばならないものを選択することからなる。次いでステップ10においてロックRDYLKは解除される。

【0052】次のステップ11は、DQSELを実行しているプロセッサのプロセッサ制御ブロックCPCB内



13

に含まれているメールボックスロックDCLKをテストすることからなる。ロックDCLKがかけられているならば(DCLK=1)、これは、プロセッサ間の対話が既に進行中であり、結果的にディスパッチングにおける対立が存在していることを意味する。そうするとマイクロプログラムは中央モジュールCTSELを呼び出す。

【0053】これとは反対にロックDCLKが解除されていれば(DCLK=0)、マイクロプログラムはステップ12においてこのロックをかけ(DCLK:=1)、処理NJPに關係するリンクのビットDを1にする。次いでステップ13が続き、選択された処理NJPの番号がブロックCPCBの対応するフィールド内に登録される。ステップ13はモジュールEXECを呼び出し、次にこれを図9を参照して説明する。

【0054】図9のフローチャートは、モジュールEXECを実行する任意のいずれかのプロセッサCPUiの一般的ケースを示す。このモジュールは、プロセッサCPUによって実行されるオペレーションDQSELによって呼び出されたときには、同じプロセッサによって実行される、即ちCPUi=CPUであることが理解される。

【0055】モジュールEXECは、ステップ14において、実行されるべき処理NJPの番号を考慮するためにプロセッサCPUiのプロセッサ制御ブロックCPCBをアクセスことから開始する。次いでステップ15において、CPUiにおいて進行中の処理CPJの内容を保存するために、この処理の処理制御ブロックPCBをアクセスする。次にステップ16において、ロックRDYLKのTEST AND SETオペレーションを実行し、次いでステップ17において、キューQ/PR/RDYをアクセスしてそれを更新する。この更新は、処理リンクCJPのビットRをゼロにセットすることからなる。更に、フィールドSTATEを“レディ、非アクティブ状態”にセットするためにCJPのブロックPCBをアクセスする。次いでステップ18においてロックRDYLKが解除される。ステップ19においては、選択された処理NJPの処理制御ブロックPCBがアクセスされ、この処理の内容がプロセッサのレジスタ内にロードされる。ステップ20において、プロセッサCPUiのブロックCPCBがアクセスされ、そのロックDCLKiが解除される。ロックRDYLKのTEST AND SETオペレーションがステップ21において再び実行され、次いでステップ22において、フィールドTENを更新し且つ処理リンクNJPのビットR及びDをそれぞれ1及び0にセットするために、キューQ/PR/RDYが再びアクセスされる。更に、NJPのブロックPCBのフィールドSTATEが“アクティブ”状態にされる。このオペレーションの後、ステップ23においてロックRDYLKは再び解除される。

【0056】Vタイプ命令によってトリガされるオペレ

14

ーションを図10のフローチャートに示す。V命令は、P命令に相補的な命令である。これは、この命令を含む処理が事象またはメッセージを特定のセマフォに通知することを意味する。

【0057】V命令を実行するためのマイクロプログラムはステップ24において、命令Vにおいて識別されたセマフォSEMと關係するロックSEMLKにおけるTEST AND SETオペレーションから開始する。ステップ25は、セマフォをアクセスし、その内容を分析し、それを更新することからなり、更新は特に、カウンタSCTを1単位だけ増分することからなる。カウンタSCTが正またはゼロであったならば、これは、命令Vによって通知された事象またはメッセージを待っている処理はないことを意味する。その場合にはステップ26においてロックSEMLKが解除され、処理の次の命令が正常に実行され得る。セマフォがメッセージを含むタイプのものであるならば、セマフォの更新は更に、メッセージリンクをメッセージキューに加えることからなる。もはや空きメッセージリンクがない特定のケースにおいては、マイクロプログラムは、この状況を取扱うために専用モジュールを呼び出す。

【0058】カウンタSCTが負であったならば、これは、少なくとも1つの処理がセマフォに關係するキューQ/PR/S内の事象またはメッセージを待っていることを意味する。次いでステップ27において、キューQ/PR/Sから取り出され且つレディ状態処理キューQ/PR/RDYに入れられねばならない処理NJPを選択するためにこのキューが分析される。このために、ステップ28においてロックRDYLKにおけるTEST AND SETオペレーションが実行され、次いでステップ29において、選択された処理リンクNJPがキューQ/PR/SからキューQ/PR/RDYにシフトされる。次いでステップ30においてロックSEMLKは解除される。

【0059】次いでステップ31においてカウンタCSTがテストされる。CSTがゼロでないならば、マイクロプログラムは中央モジュールCTSELにジャンプする。これとは反対にカウンタCSTがゼロであるならば、マイクロプログラムは、処理をレディ状態処理キューQ/PR/RDYに加える状況を取扱うために専用モジュールEQSELを呼び出す。

【0060】モジュールEQSELの実行はステップ33において、新たな処理NJPがアクティブになる必要があるか判断し、もしそうならば、プロセッサCPU、CPUiのうちのどれがそれを実行すべきかを選択するために、レディ状態処理キューQ/PR/RDYをアクセスする処理を行なう。このオペレーションは、ポイントが最低プライオリティを有する最後のアクティブ処理のリンクにあるならば極めて単純なものとなり得る。その場合には、NJPのプライオリティレベルとポイント

15

によって示されている処理のそれとを比較すれば十分である。NJPが同じかまたはより低いプライオリティを有するならば、それは選択されない。これとは反対にNJPがより高いプライオリティを有するならば、NJPはこの処理に置き換わらねばならず、この処理のリンクのフィールドTENによって問題のプロセッサが示される。

【0061】処理NJPが選択されなかったならば、オペレーションはステップ34においてロックRDYLKを解除して終了する。

【0062】これとは反対に処理NJPが選択されたならば、システムのプロセッサCPU、CPUiの1つは、進行中の処理を選択された処理NJPと置き換えるために、その実行に割込む必要がある。選択されたプロセッサが、モジュールEQSELを実行しているプロセッサCPUであったならば、そのときはステップ35においてロックRDYLKを解除してオペレーションは続行され、次いでステップ36においてプロセッサCPUのメールボックスのロックDCLKがテストされる。ロックがかけられているならば、マイクロプログラムは中央モジュールCTSELにジャンプする。これとは反対にロックDCLKが解除されているならば、ステップ37においてマイクロプログラムはこのロックをかけ、ステップ38において、選択された処理NJPの番号をプロセッサCPUのブロックCPCBの対応するフィールド内に登録する処理に進む。ステップ38は、図9を参照して説明されるモジュールEXECを呼び出して終了する。

【0063】ステップ33において行われた分析によって別のプロセッサCPUiが選択されたならば、ステップ39においてロックRDYLKは解除され、マイクロプログラムは図11に示したオペレーションAを続ける。

【0064】次のステップ40は、対話ロックMPDLのテストを実行する。このロックがかけられているならば、マイクロプログラムは中央モジュールCTSELにジャンプする。これとは反対にMPDLが解除されているならば、ステップ41においてマイクロプログラムはこのロックをかける。次にステップ42において、マイクロプログラムは、選択されたプロセッサCPUiのメールボックスロックDCLKiをテストする。ロックDCLKiがかけられているならば、マイクロプログラムはステップ43においてMPDLを解除した後に、中央モジュールCTSELにジャンプする。これとは反対にDCLKiが解除されているならば、マイクロプログラムはステップ44においてロックをかける。次いでステップ45へ進み、選択された処理NJPの番号を、選択されたプロセッサCPUiのブロックCPCBの対応するフィールド内に登録する。

【0065】次いでマイクロプログラムはステップ46

16

において、割込みEXECiをプロセッサCPUiに送ることによりプロセッサ間の対話手段を使用する。次にステップ47においてマイクロプログラムは、プロセッサCPUiによって送られるべき肯定応答を待つ状態に入る。一旦この肯定応答を受け取ると、ステップ48において対話ロックMPDLは解除され、これでオペレーションは終了する。そうするとプロセッサは、それに割り当てられた処理の実行を継続することができる。

【0066】ステップ46において割込みEXECiがプロセッサCPUiに送られると、このプロセッサにおいて割込みマネージャが能動化され、割込みマネージャは図9を参照して既に説明したモジュールEXECを呼び出す。従って、メールボックスロックDCLKiが解除されるのは、ステップ20においてこのモジュールが実行されている間だけである。

【0067】上記実施例の大きな特徴は、プロセッサにおけるこの同じプロセッサによるモジュールEXECの呼び出し(図8のステップ13及び図10のステップ38)が、プロセッサがそれ自体にメッセージを通知し得る前出の仏国特許出願第90、06948号に記載の汎用割込み機構によって行われるようになっていることである。このような場合には、対話ロックMPDLは関与しないことが理解される。

【0068】オペレーションを更に最適化するためには、ブロックSCBのフィールドCSTが対立インジケータを含むようにすることができ、この対立インジケータは、ロックDCLK及びMPDLのテスト(ステップ11、37、40)の際に対立が検出されるごとに0でない状態におかれる。即ち、CSTのテスト(ステップ7、31)によって、ディスパッチングオペレーション間のいかなる対立の存在も極めて迅速に認識される。

【0069】上述のセマフォによるオペレーションP及びVの詳細な説明は、ディスパッチングの分散実行を可能するため専用モジュールEQSEL及びDQSELを中央モジュールCTSELとどのように協働させて使用して、並行処理の可能性を増大するかを示している。特に、複数のプロセッサが異なるディスパッチングオペレーションの所定のフェーズを同時に実行でき、唯一の制約条件は、所定の共有資源(セマフォ、キューQ/PR/RDY、対話機構、メールボックス)への同時アクセスが禁止されていることであることに留意されたい。かかる禁止は、かかる資源に関係するロックによって極めて単純に保証され、従ってシステムの保全性を保証する。

【0070】当然ながらモジュールEQSEL及びDQSELは、命令P及びV以外の命令にも応答して介在し得る。これは特に、その機能は新たな処理をシステムに導入することである命令STARTにも当てはまる。

【0071】本発明によれば、システムオペレーションの内容に応じて選択される他の専用モジュールを備える

17

こともできる。例えば、命令が処理リンクのキューQ/PR/RDYへのシフトを指示するケースに対してはモジュールCHSELKを備えることができる。これは例えば、それを含む処理のプライオリティレベルを変更する命令CHOPや、処理の実行権を与える結果となる命令RLQにも当てはまる。

【0072】これらの他のケースを扱うマイクロプログラムは、前述のモジュールEQSEL及びDQSELの詳細な説明を利用することにより実現することができる。当業者はこれらを独自に実現し得るであろうし、更なる説明は必要ないであろう。

【図面の簡単な説明】

【図1】本発明を実施するための情報処理システムの中央サブシステムを示す。

【図2】ディスパッチングオペレーションに使用される処理制御ブロックのフォーマットを示す。

【図3】プロセッサ制御ブロックのフォーマットを示す。

【図4】種々の処理キューに使用される処理リンクのフォーマットを示す。

【図5】セマフォのフォーマットを示す。

【図6】セマフォに関係するメッセージキューを形成す

18

る役割を果たすメッセージリンクのフォーマットを示す。

【図7】ディスパッチングオペレーションの際に使用されるシステムデータを含むシステム制御ブロックのフォーマットを示す。

【図8】セマフォによるPタイプ命令の実行示すフローチャートである。

【図9】マイクロプログラムEXECのフローチャートである。

【図10】セマフォによるVタイプオペレーションのフローチャートを示す。

【図11】セマフォによるVタイプオペレーションのフローチャートの続きを示す。

【符号の説明】

CPU、CPU1 プロセッサ

DTS データバス

ADS アドレスバス

CDS 制御ライン

SB システムバス

IOU 入/出力装置

MU 中央記憶装置

【図3】

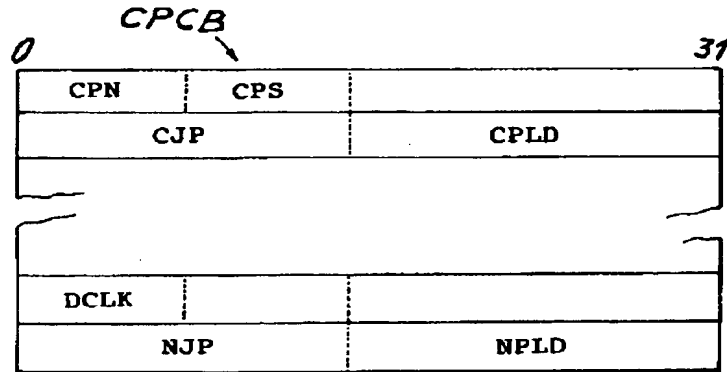


FIG. 3

【図4】

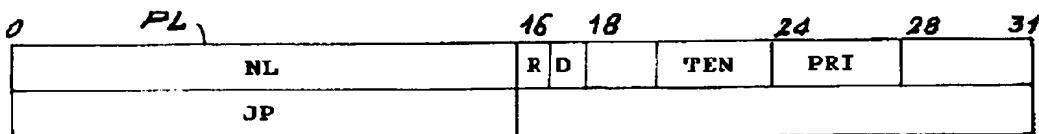
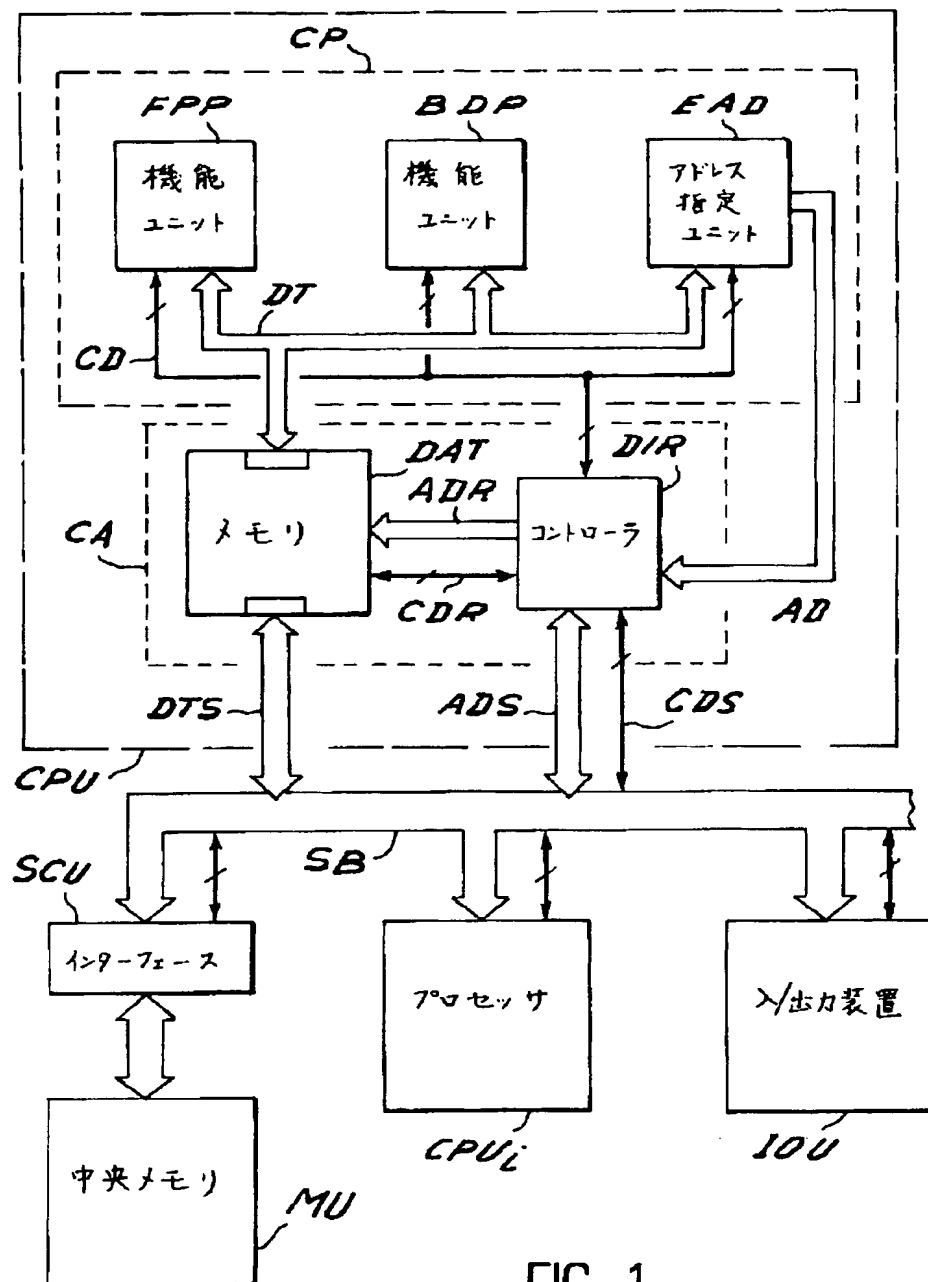


FIG. 4

【图 1】



【図2】

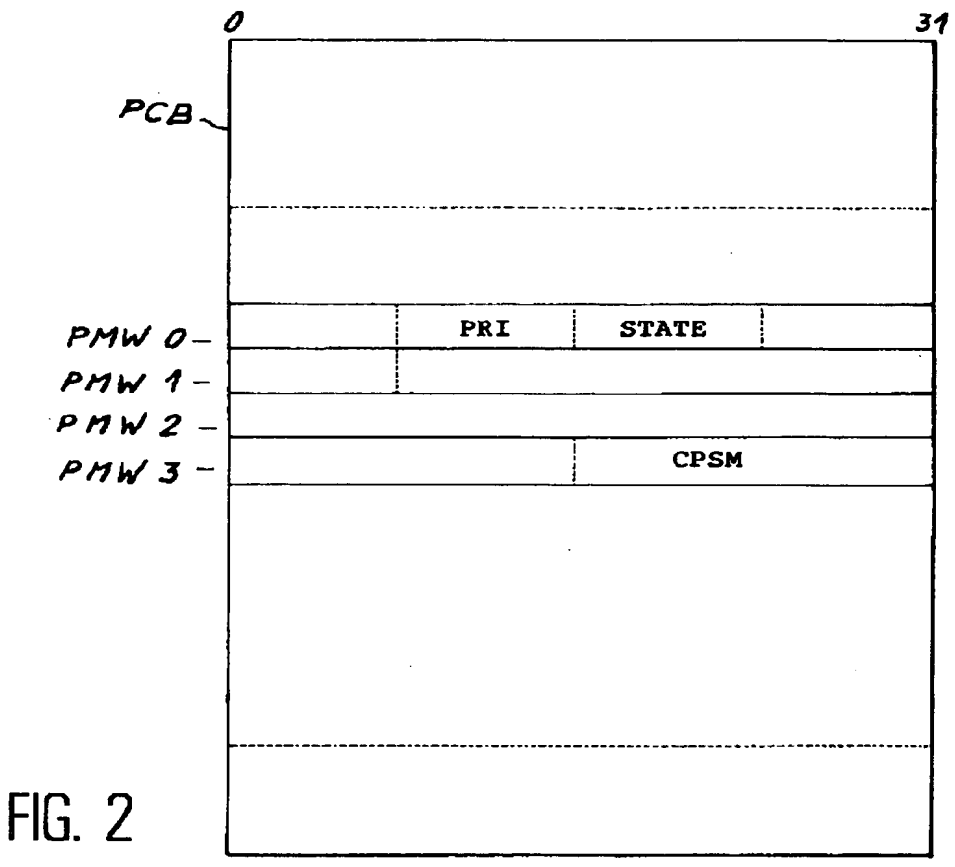


FIG. 2

【図5】

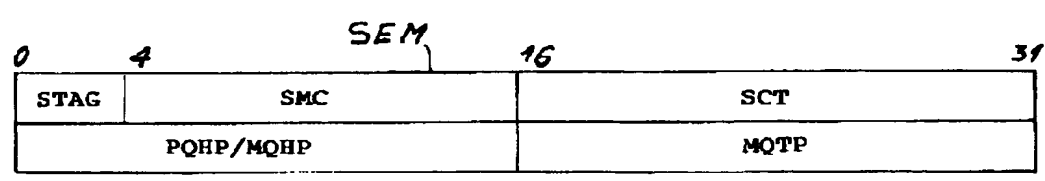


FIG. 5

【図6】

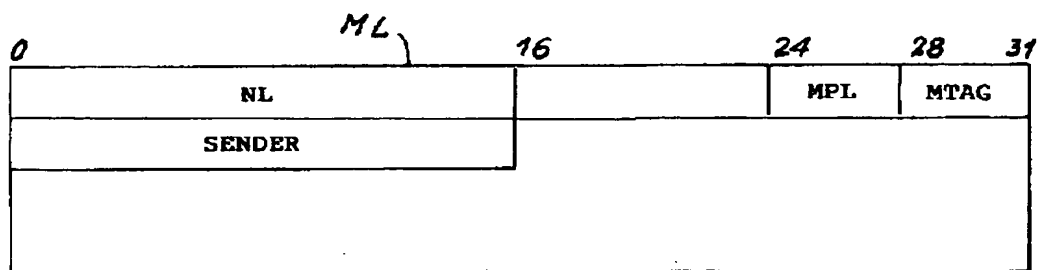


FIG. 6

【図7】

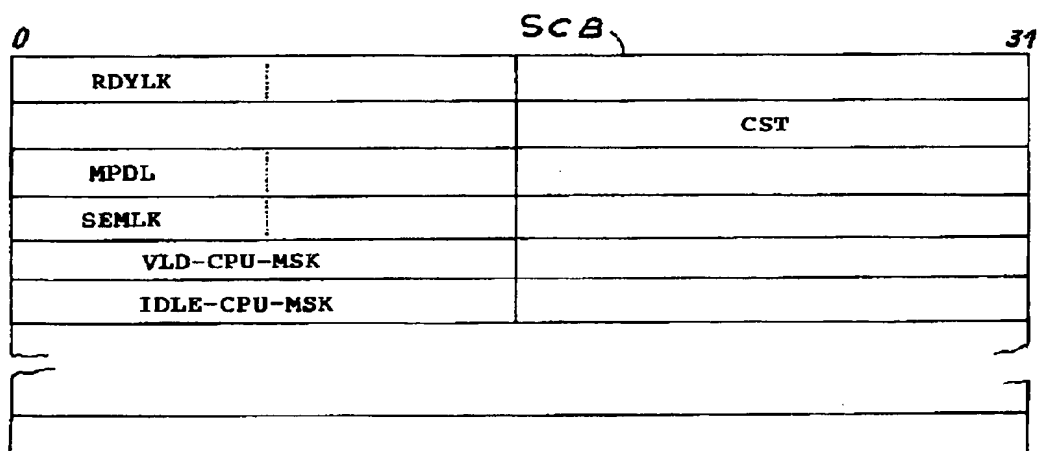


FIG. 7

【図8】

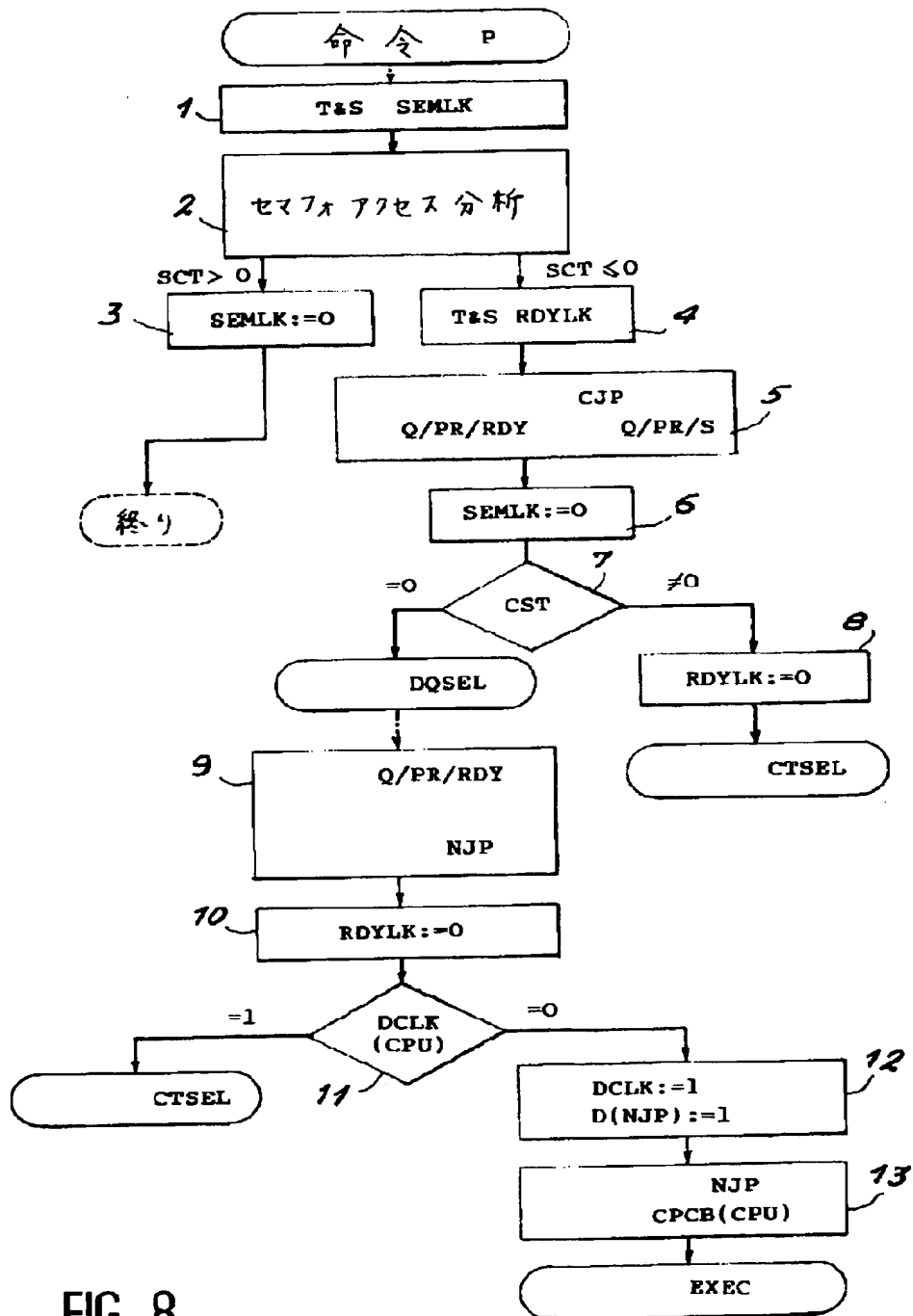


FIG. 8

【図9】

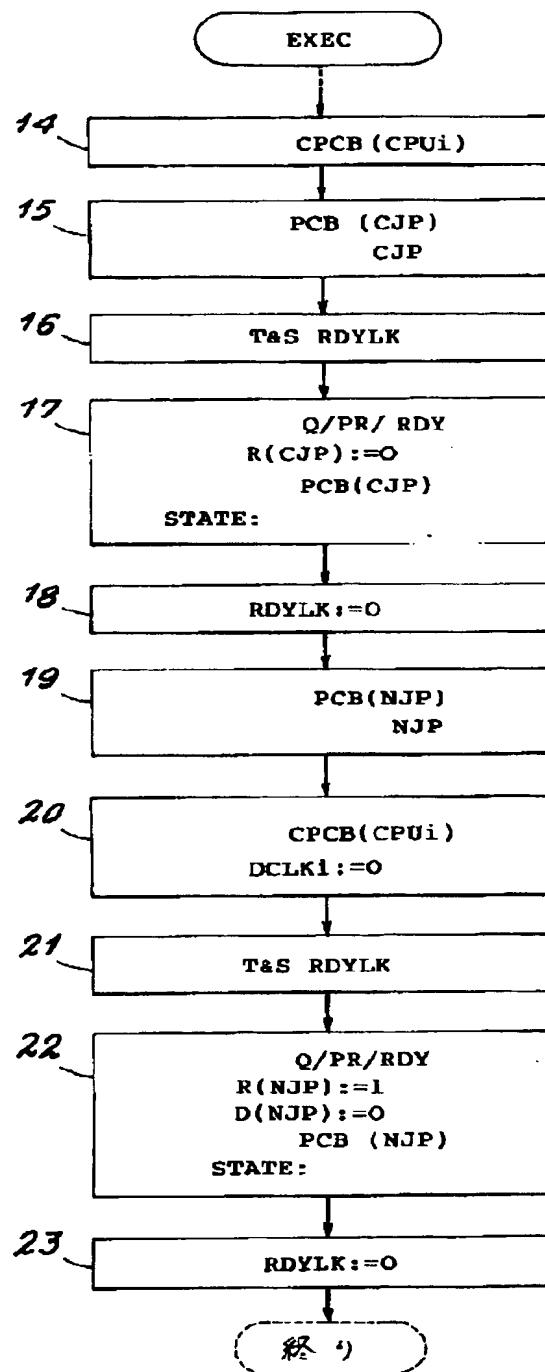


FIG. 9



【図10】

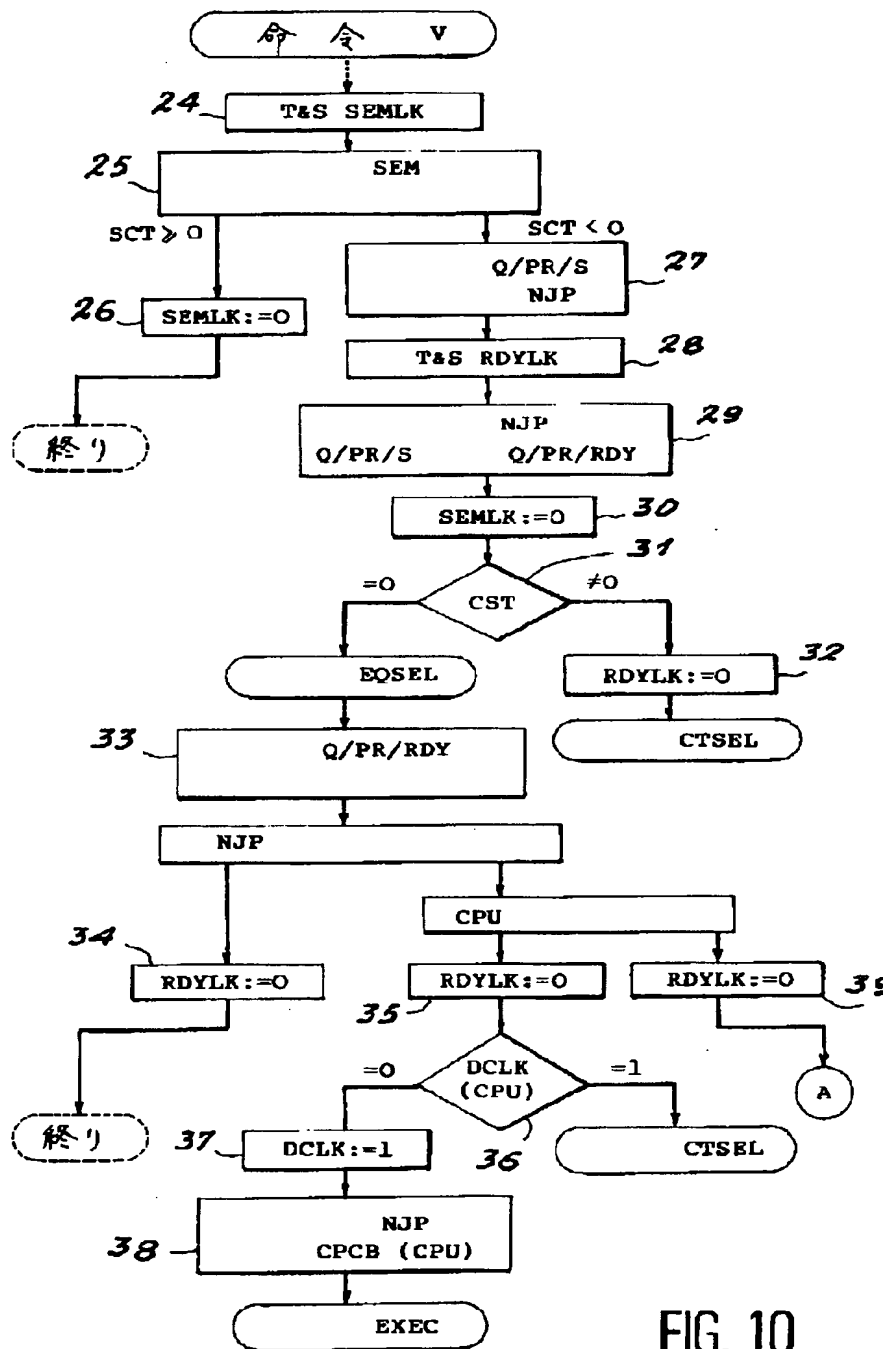


FIG. 10

【図11】

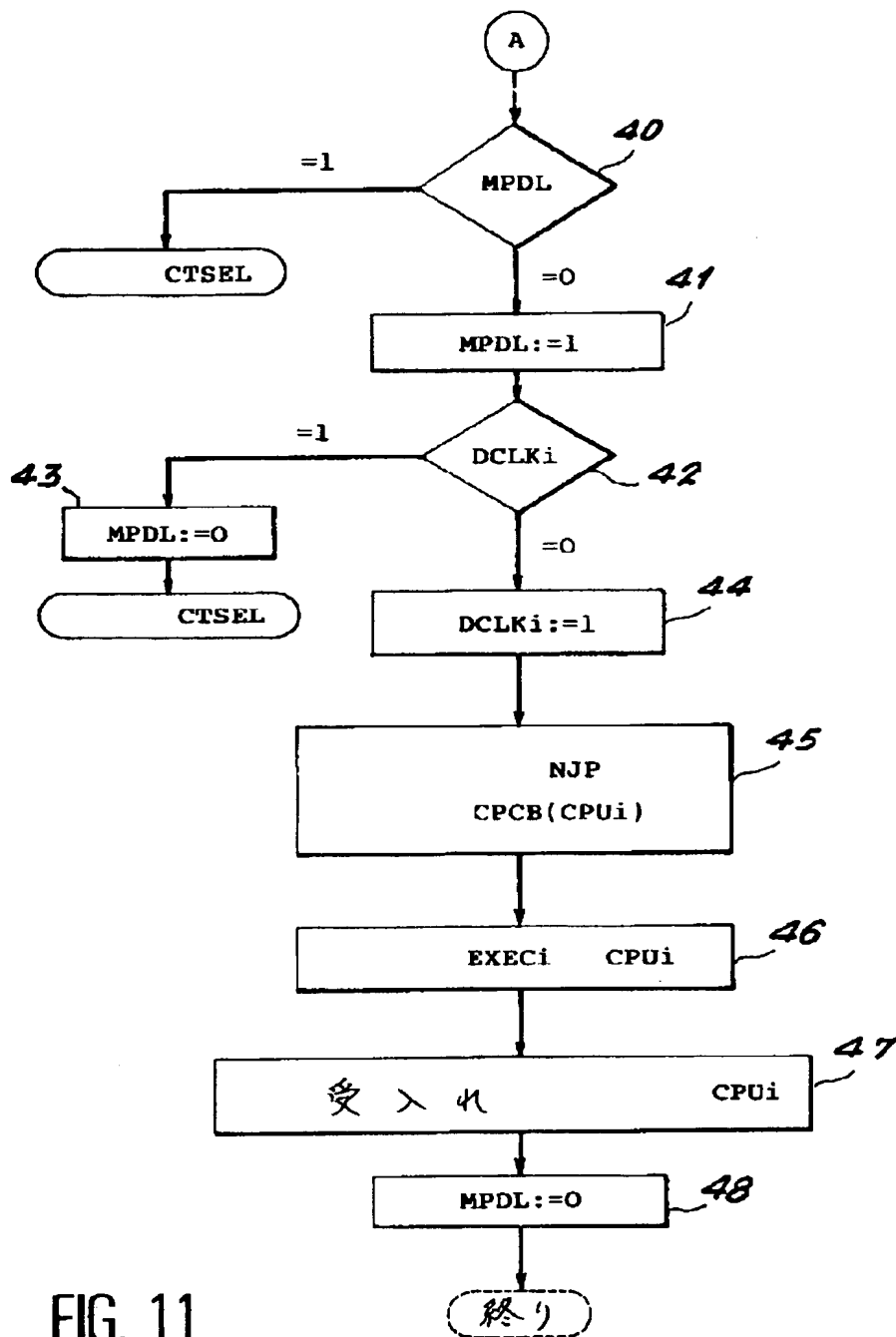


FIG. 11